

Deteksi Fraud Menggunakan Metode Model Markov Tersembunyi pada Proses Bisnis

Andrean Hutama Koosasi, Riyanarto Sarno, dan Abdul Munif

Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember (ITS)

Jl. Arief Rahman Hakim, Surabaya 60111 Indonesia

e-mail: riyanarto@if.its.ac.id

Abstrak—Model Markov Tersembunyi merupakan sebuah metode statistik berdasarkan Model Markov sederhana yang memodelkan sistem serta membaginya dalam 2 (dua) state, state tersembunyi dan state observasi. Dalam pengerjaan studi ini, penulis mengusulkan penggunaan metode Model Markov Tersembunyi untuk menemukan fraud didalam sebuah pelaksanaan proses bisnis. Dengan penggunaan metode Model Markov Tersembunyi ini, maka pengamatan terhadap elemen penyusun sebuah kasus/kejadian, yakni beberapa aktivitas, akan diperoleh sebuah nilai peluang, yang sekaligus memberikan prediksi terhadap kasus/kejadian tersebut, sebuah fraud atau tidak. Hasil eksperimen ini menunjukkan bahwa metode yang diusulkan mampu memberikan prediksi akhir dengan evaluasi TPR sebesar 87,5% dan TNR sebesar 99,4%.

Kata Kunci—Fraud, Model Markov Tersembunyi, Proses Bisnis, Probabilitas, SOP.

I. PENDAHULUAN

METODE Model Markov Tersembunyi merupakan sebuah metode untuk merepresentasikan atau menampilkan distribusi probabilitas/kemungkinan berdasarkan sejumlah data observasi yang ada [1]. Model Markov Tersembunyi merupakan metode pengembangan dari Teorema Bayes (Aturan Bayes) dan Proses Markov.

Penelitian ini mengangkat implementasi metode Model Markov Tersembunyi untuk penggalan *fraud* pada proses bisnis. Proses bisnis merupakan sekumpulan aktivitas bisnis yang saling berkaitan secara logis dan bekerjasama untuk menghasilkan sebuah nilai/tujuan tertentu [2, 3]. Alur kerja proses bisnis ialah sebagai berikut: pembentukan Prosedur Operasional Standar (SOP) sebagai tatanan baku sebuah proses bisnis yang akan dilakukan, akan diikuti dengan beberapa kasus yang menjalankan proses bisnis tersebut, sesuai dengan SOP yang ada. Faktanya, kasus-kasus yang berjalan di lapangan seringkali tidak sesuai/tidak cocok seutuhnya dengan SOP yang telah dibuat. Ketidaksesuaian/ketidaccocokan ini cenderung akan mengarah kepada terbentuknya pelanggaran/*fraud*.

Fraud tidak bisa dihindari, walaupun memiliki peluang kemunculan yang sangat kecil dibandingkan dengan sejumlah besar kasus yang berjalan [4]. Dengan mempertimbangkan beberapa hal, seperti tingkat akurasi yang terbilang kurang baik untuk mencari *fraud* yang spesifik, dan hendak mengembangkan keberagaman metode penggalan *fraud*, maka penelitian ini mengajukan metode Model Markov Tersembunyi sebagai metode alternatif mencari *fraud* pada

event log yang didasarkan dari SOP proses bisnis yang telah ada.

Model Markov Tersembunyi ialah metode gabungan dari beberapa algoritma dasar, seperti teorema Bayes, algoritma Viterbi [5], dan algoritma Baum-Welch (*Forward-Backward Algorithm*) [6]. Tingkat pengukuran kualitas metode ini akan ditentukan dari hasil akurasi (*accuracy*), *precision*, sensitivitas (TPR), dan spesifisitas (TNR) [7]. Hasil akhir yang diharapkan menjadi keluaran dari penelitian ini ialah pengelompokan kelas *state* tiap kasus dengan baik, diikuti dengan evaluasi kualitas metode yang baik juga dari segi persentase angka.

II. DASAR TEORI

A. Fraud

Fraud merupakan penyimpangan yang merugikan, baik dalam skala kecil maupun besar [4]. Jenis-jenis *fraud* yang terjadi menurut proses bisnis dapat dipahami sebagai *process-based fraud* (PBF) [2]. Jenis-jenis PBF yang dimaksud akan dijelaskan sebagai berikut.

1) Skipped Activity

Terjadi dengan cara pelaku melompati/melewati aktivitas yang bersifat *sequence*/sekuens biasa yang menjadi urutan baku pada SOP dari proses bisnis yang dijalankan tersebut [2].

2) Wrong Throughput Time

Terjadi dengan cara menyalahi alokasi waktu yang telah ditentukan oleh SOP [2].

3) Wrong Resource

Terjadi apabila sumber daya/*resource* mengerjakan aktivitas yang tidak dialokasikan untuk diselesaikan oleh sumber daya tersebut menurut SOP [2].

4) Wrong Duty

Terjadi apabila pelaku/*resource* mengerjakan lebih dari satu aktivitas dalam sebuah kasus, bisa juga disebabkan oleh penyelesaian aktivitas oleh pelaku dengan wewenang jabatan lebih rendah [2].

5) Wrong Pattern

Terjadi dengan cara pelaku menjalankan proses bisnis tidak sesuai dengan urutan aktivitas pada SOP [4].

6) Wrong Decision

Terjadi dengan cara pelaku mengambil keputusan/*decision* yang salah [4].

B. Algoritma Viterbi

Algoritma Viterbi merupakan algoritma yang digunakan untuk mencari kejadian selanjutnya/aktivitas selanjutnya

dengan melakukan pencarian kejadian/aktivitas yang memiliki nilai total probabilitas tertinggi [5, 9]. Dalam kaitannya dengan Model Markov Tersembunyi, algoritma Viterbi berperan dalam penyelesaian problem *decoding*. Tugas algoritma ini selaku *decoder* ialah mencari *hidden path* terbaik, seperti telah dijabarkan pada definisi sebelumnya. Apabila diketahui sejumlah observasi $O = o_1, o_2, o_3, \dots, o_T$ dan *state* obyek $Q = q_1 q_2 q_3 \dots q_T$, maka urutan sekuens Viterbi/Viterbi path dapat dibentuk dengan menggunakan (2).

$$v_t(j) = \max_{q_0, q_1, \dots, q_{t-1}} P(q_0, \dots, q_{t-1}, o_1 \dots o_t, q_t = j | \lambda) \quad (2)$$

Dengan λ menyatakan bentuk HMM $\lambda = (A, B)$. Untuk menghitung nilai probabilitas Viterbi jika diberikan *state* q_j dan waktu t , maka nilai $v_t(j)$ dapat dirumuskan sesuai (3).

$$v_t(j) = \max_{i=1}^N v_{t-1}(i) a_{ij} b_j(o_t) \quad (3)$$

Dengan $v_{t-1}(i)$ menyatakan probabilitas Viterbi path sebelumnya, a_{ij} menyatakan probabilitas transisi dari *state* sebelumnya q_i menuju *state* sekarang q_j , dan $b_j(o_t)$ menyatakan kesamaan observasi antar *state* o_t pada *state* sekarang (j) [9].

C. Algoritma Forward

Algoritma *Forward* merupakan algoritma yang menghitung probabilitas kemungkinan pada waktu t , diberikan kejadian yang telah lampau [10]. Jika diberikan persamaan HMM $\lambda = (A, B, \pi)$, maka rumusan algoritma *Forward* untuk menghitung $P(O|\lambda) = \alpha(t, i)$ ditunjukkan pada Persamaan 4 berikut.

$$\alpha(t, i) = P(O_1, O_2, O_3, \dots, O_t, q_t = S_i) \quad (4)$$

Dengan $\alpha(t, i)$ menyatakan nilai yang dihasilkan oleh persamaan algoritma *Forward* pada waktu t dan *state* ke- i , dan $P(O_1, O_2, O_3, \dots, O_t, q_t = S_i)$ menyatakan sejumlah nilai probabilitas dari rangkaian pengamatan $O_1, O_2, O_3, \dots, O_t$ pada *state* ke- i dan waktu t .

D. Algoritma Backward

Algoritma *Backward* merupakan algoritma yang memiliki cara kerja sama dengan algoritma *Forward*, namun memiliki titik awal (*start position*) dari akhir proses [10]. Rumusan algoritma *Backward* untuk menghitung $P(O|\lambda) = \beta(t, i)$ ditunjukkan pada (5) berikut.

$$\beta(t, i) = P(O_{t+1}, O_{t+2}, \dots, O_T | q_t = S_i) \quad (5)$$

Dengan $\beta(t, i)$ menyatakan nilai yang dihasilkan oleh persamaan algoritma *Backward* pada waktu t dan *state* ke- i , dan $P(O_{t+1}, O_{t+2}, \dots, O_T, q_t = S_i)$ menyatakan nilai probabilitas dari pengamatan $O_1, O_2, O_3, \dots, O_t$ pada *state* ke- i dan waktu t .

E. Algoritma Baum Welch

Algoritma Baum-Welch merupakan algoritma gabungan dari algoritma *Forward* (α) dan algoritma *Backward* (β). Algoritma ini digunakan untuk melakukan penggalan terhadap parameter/atribut penyusun metode Model Markov Tersembunyi [6]. Desain algoritma Baum-Welch memungkinkan untuk memaksimalkan observasi terhadap rangkaian pengamatan $P(O_1, O_2, O_3, \dots, O_n)$. Bentuk umum Baum-Welch dapat dilakukan ekspansi sejumlah n sekuens berbeda berisikan $m_1, m_2, m_3, \dots, m_n$, seperti pada (6).

$$\begin{aligned} O^1 &: O_1^1, O_2^1, O_3^1, \dots, O_{m_1}^1 \\ O^2 &: O_1^2, O_2^2, O_3^2, \dots, O_{m_2}^2 \\ &\dots \end{aligned} \quad (6)$$

$$O^n : O_1^n, O_2^n, O_3^n, \dots, O_{m_n}^n$$

Sedangkan untuk penyelesaian algoritma Baum-Welch dapat dilihat pada (7).

$$P(q_t = k, q_{t+1} = l | O, \lambda) := \frac{\alpha(t, k) a_{kl} b_l(O_{t+1}) \beta(t+1, l)}{P(O | \lambda)} \quad (7)$$

Dengan q_t menyatakan sekuens pada waktu t , q_{t+1} menyatakan sekuens pada waktu $t+1$, O menyatakan observasi yang didasarkan pada k (q_t) dan l (q_{t+1}), α menyatakan nilai solusi dari algoritma *forward*, dan β menyatakan nilai solusi dari algoritma *backward*.

F. Proses Markov

Proses Markov ialah kumpulan variabel dependen yang tersusun menjadi sebuah proses, yang memenuhi keseluruhan atribut syarat dari properti Markov (*Markov properties*) [11]. Properti Markov terdiri dari:

1. Proses Markov memiliki matriks distribusi probabilitas, yakni probabilitas awal dan probabilitas transisi [9].
2. Pada sebuah sekuens $(x_1, x_2, \dots, x_{n-1})$, prediksi terhadap nilai selanjutnya (x_n) hanya akan bergantung pada nilai sebelumnya saja (x_{n-1}) [9].

G. Model Markov Tersembunyi (HMM)

Model Markov Tersembunyi merupakan sebuah model stokastik berganda yang memiliki dua tingkat hirarki (*state*) [10]. *State* yang dimaksud terdiri dari *state* yang dapat diamati (*observable state*) dan *state* yang bersifat tersembunyi dari pengguna, dan bersifat hasil dari observasi (*hidden state*) [11].

Bentuk lengkap dari persamaan Model Markov Tersembunyi ini seperti dituliskan pada (8).

$$HMM \lambda = (A, B, \pi) \quad (8)$$

Dengan A menyatakan matriks distribusi probabilitas *state*, B menyatakan matriks distribusi probabilitas observasi, dan π menyatakan vektor probabilitas awal (*initial probabilities*) [5]. Pembentukan vektor probabilitas awal memperhatikan posisi *start* awal sebuah proses bisnis dimulai pada *petri net*. Adapun rumusan pembentukan vektor probabilitas awal seperti pada (9).

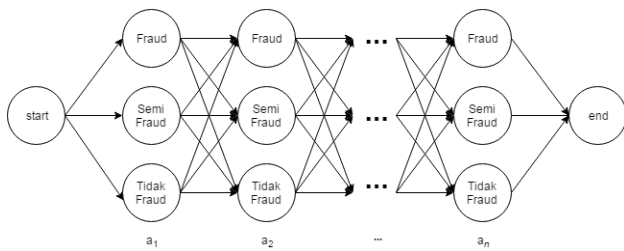
$$\pi_i = P(q_1 = S_i) \quad (9)$$

Pembentukan matriks distribusi probabilitas *state* $A = [a_{ij}]$ seperti dijelaskan pada (10).

$$a_{ij} = P(q_{t+1} = S_j | q_t = S_i) \quad (10)$$

Pembentukan matriks distribusi probabilitas observasi $B = [b_j(k)]$ seperti dijelaskan pada (11).

$$b_j(k) = P(V_k | S_j) \quad (11)$$



Gambar 1. Diagram Skema Modifikasi Baum-Welch

Tabel 1.
Pengaturan Posisi Atribut PBF

Atribut 1	Atribut 2	Atribut 3	Atribut 4	Atribut 5
WDutyS	SkipSeq	WRes	WDutyD	WDutyC
Atribut 6	Atribut 7	Atribut 8	Atribut 9	Atribut 10
WPattern	Tmax	Tmin	SkipDec	WDec

Rangkaian matriks dan vektor penyusun parameter Model Markov Tersembunyi diatas memerlukan sejumlah t sekuens observasi, sehingga $O = O_1, O_2, O_3, \dots, O_t$, dimana O_t merupakan bagian dari V , dan sejumlah R pengamatan atas sekuens observasi tersebut.

III. METODOLOGI

Pada bab ini akan dijelaskan mengenai metodologi pemecahan masalah yang dihadapi pada penelitian ini.

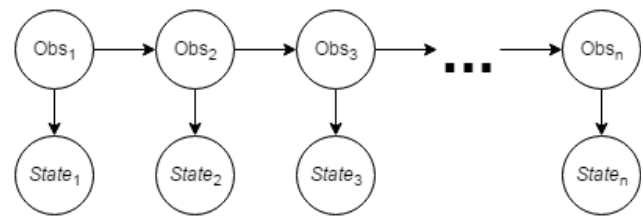
A. Dataset

Masukan dari metode ini ialah sekumpulan data yang tersusun menjadi *event log*, yang telah dikembangkan dari SOP yang ada. SOP yang dimaksudkan merupakan simulasi peminjaman kredit suatu bank. Dataset *event log* yang digunakan untuk proses *training* berjumlah 1000 data, dengan rincian 20 *case fraud*, 14 *case semi fraud* dan sisanya *case tidak fraud*. Sedangkan untuk proses *testing* terdiri dari 200 data dengan rincian 5 *case fraud*, 3 *case semi fraud*, dan sisanya *case tidak fraud*. Ketentuan konten dataset *event log* yang digunakan, baik untuk proses training maupun testing, ialah sebagai berikut: kolom pertama merupakan kolom nomor *case* kesekian (*case_ID*), kolom kedua hingga kesebelas berisi 10 atribut PBF, dengan diikuti jumlah pelanggaran atribut tersebut untuk masing-masing *case*, kolom kedobelabel merupakan status riil dari tiap *case*, dan kolom ketigabelabel yang nantinya akan diisi oleh hasil prediksi metode ini.

B. Pembuatan Model Markov Tersembunyi

Pemrosesan *event log* pertama ini menggunakan metode Model Markov Tersembunyi. Pemodelan terhadap bentuk ini memerlukan *hidden state* dan *observable state*. Observasi akan diarahkan pada sekuens atribut pelanggaran *event log*, sementara *hidden* akan diarahkan kepada *state* pelanggaran setiap atribut untuk tiap kasus ($State_1, State_2, \dots, State_n$).

Model Markov Tersembunyi yang digunakan akan dibantu dengan algoritma Baum-Welch yang telah dijelaskan pada bab dasar teori. Diagram pencarian probabilitas yang dibentuk sesuai dengan algoritma Baum-Welch hasil adaptasi/modifikasi terhadap inisialisasi *state* pada paragraf sebelumnya ditunjukkan pada Gambar 1, dimana pada setiap atribut ke n (a_n) terdapat probabilitas ketiga *state* yang harus dihitung (*fraud*, *semi fraud*, *tidak fraud*).



Gambar 2. Diagram Trellis untuk Pencarian Probabilitas

Tabel 2.
Matriks Probabilitas Awal (π)

	Fraud	Semi fraud	Tidak fraud
Start	0,02	0,014	0,966

Tabel 3.
Matriks Transisi Antar-State

	Fraud	Semi fraud	Tidak fraud
Fraud	0,022	0,022	0,956
Semi fraud	0,333	0,333	0,333
Tidak fraud	0,019	0,001	0,979

Diagram trellis pada Gambar 2 merupakan bentuk jadi yang menggambarkan cara memperoleh nilai probabilitas dari iterasi perkalian metode Model Markov Tersembunyi. Pada Gambar 2, posisi *hidden state* pada rangkaian ialah *state* per kasus, dan observasi berasal dari atribut PBF.

C. Pengaturan Posisi Atribut PBF

Pengaturan posisi atribut diperlukan untuk mencari kombinasi observasi antar-atribut yang optimal. Kombinasi observasi yang tepat diperlukan supaya pembentukan matriks transisi antar-state menghasilkan nilai maksimal, yang mempengaruhi pada hasil pengamatan *hidden state* berdasarkan observasi dari posisi atribut yang telah diatur sebelumnya.

Pengaturan ulang posisi atribut ini melihat nilai linguistik yang telah didefinisikan sebagai pendapat pakar pada penelitian sebelumnya [2]. Nilai linguistik memiliki besaran yang rancu, sehingga dilakukan konversi kedalam nilai/skala numerik. Dari keempat data pendapat pakar yang diambil, terdapat lima variasi skala linguistik yang digunakan sehingga konversi ke skala numerik ini terdiri dari skala 1-5 (semakin tinggi semakin penting). Kemudian setelah konversi kedalam bentuk numerik selesai, diambillah nilai *mean* (rerata) dari keempat pakar, dan menghasilkan pengaturan posisi atribut yang baru seperti ditunjukkan pada Tabel 1.

D. Pembuatan Matriks Probabilitas State

Matriks probabilitas dibuat dari pengamatan terhadap *event log* yang menjadi dataset masukan. Penentuan parameter HMM memanfaatkan algoritma Baum-Welch. Tahap pertama ialah melakukan inisialisasi terhadap matriks/vektor probabilitas awal (*start value*). Inisialisasi matriks ini berlangsung untuk semua *state*. Nilai dari masing-masing inisialisasi awal *state* diambil dari perbandingan antara frekuensi atribut bernilai *state* tersebut dengan keseluruhan jumlah kasus dalam dataset. Inisialisasi matriks probabilitas awal seperti ditunjukkan pada Tabel 2.

Tabel 4.
Potongan Matriks Emisi Atribut-State

	a_1	no- a_1	a_2	no- a_2	a_3	no- a_3
Fraud	0,0050	0,0950	0,0900	0,0100	0,0150	0,0850
Semi fraud	0,0000	0,1000	0,0000	0,1000	0,0214	0,0786
Tidak fraud	0,0042	0,0958	0,0000	0,1000	0,0019	0,0981

Berikutnya dilakukan inisialisasi matriks transisi antar-*state* berisi nilai probabilitas antara *state n* dengan *state n+1*. Modifikasi dilakukan pada tahap ini, dimana matriks transisi antar-*state* dipisahkan sebanyak *n* transisi. Contoh bentukan matriks transisi antar *state* untuk *state a₁* ke *state a₂* dapat dilihat pada Tabel 3.

Matriks transisi antar *state* pada Tabel 3 berlaku sebagai distribusi probabilitas *state* (A). Jenis *state* pada baris berlaku sebagai *state ke-n*, sedangkan jenis *state* pada kolom berlaku sebagai *state ke-n+1*.

Dari jenis-jenis atribut PBF pada *event log*, dapat dibentuk matriks probabilitas kemunculan suatu atribut pada masing-masing *state* yang ditentukan. Matriks ini kemudian disebut sebagai matriks probabilitas atribut dan berlaku sebagai nilai emisi/keluaran (*emission matrix*) pada sebuah *state* yang diamati. Pemrosesan pembuatan matriks emisi lebih lanjut dijelaskan pada tahap selanjutnya, yakni penambahan kondisi no-atribut untuk penyesuaian dengan kondisi dataset.

E. Penambahan Kondisi no-Atribut

Pada *event log* yang menjadi dataset, terdapat frekuensi 0 pada setiap kolom atribut PBF untuk menandakan tidak terjadi pelanggaran pada atribut yang dimaksud. Kondisi ini dinamakan no-atribut. Pada Tabel 4 pembagian nilai probabilitas masing-masing atribut telah disertai dengan no-atribut, yang menandakan nilai probabilitas tidak terjadi pada *state* yang dimaksud. Kondisi no-atribut ini ditambahkan untuk menghindarkan nilai probabilitas 0 pada saat dilakukan penggalan *maximum path* tiap atribut dengan menggunakan algoritma Viterbi, seperti yang akan dijelaskan setelah ini.

F. Pencarian Probabilitas Per Kasus (case)

Perhitungan probabilitas antar kasus berasal dari perhitungan probabilitas antar observasi, yang menghasilkan nilai *state* dari masing-masing observasi tersebut. Pengerjaannya memanfaatkan algoritma Baum-Welch, selain untuk melakukan estimasi parameter yang tidak diketahui. Rumusan penggalan probabilitas awal (sebagai awal rantai) seperti ditulis pada (12).

$$\alpha_i(t) = \pi_i * emit_prob_{a(s)} \quad (12)$$

Dengan π_i merupakan vektor/matriks probabilitas awal, yang hanya dipakai pada *state* mula-mula, dan *emit_prob_{ij}* merupakan matriks B berisi matriks probabilitas emisi antara *observable* (atribut) *a* dengan *state s*.

Penggalan nilai probabilitas terhadap elemen sekuens selanjutnya (*t+1*) dilakukan dengan mengganti vektor probabilitas awal menjadi nilai probabilitas yang dihasilkan dari perhitungan nilai probabilitas awal. Nilai probabilitas ini juga mulai memperhitungkan dari seluruh tabel A yang ada. Rumusan nilai probabilitas pada *t+1* ditulis pada (13)

$$\alpha_i(t) = \alpha_i(t-1) * trans_prob_{ij} * emit_prob_{a(s)} \quad (13)$$

Tabel 5.
Confusion Matrix Hasil Testing Data

		Prediksi		
		Fraud	Semi fraud	Tidak fraud
Riil	Fraud	5	0	0
	Semi fraud	0	2	1
	Tidak fraud	0	1	191

Dengan $\alpha_i(t-1)$ merupakan nilai probabilitas sebelumnya, *trans_prob_{ij}* sebagai matriks A yang berisi nilai probabilitas transisi dari *state i* ke *state j*, dan *emit_prob_{ij}* merupakan matriks B berisi matriks probabilitas emisi antara *observable* (atribut) *a* dengan *state s*.

IV. HASIL

Pengujian yang dilakukan menghasilkan evaluasi disajikan dalam bentuk *confusion matrix* pada Tabel 5, untuk kemudahan menampilkan frekuensi kesalahan prediksi yang dilakukan. Kolom pada *confusion matrix* merupakan kelas hasil prediksi metode, sedangkan baris merupakan kelas riil data.

Dengan hasil yang telah ditampilkan pada Tabel 5, maka persentase evaluasi kualitas metode yang dihasilkan sebagai berikut: akurasi sebesar 99%, sensitivitas sebesar 87,5%, dan spesifisitas sebesar 99,48%.

Akurasi diperoleh dari pembagian antara frekuensi data yang diprediksi secara tepat dengan keseluruhan jumlah data. Sensitivitas merupakan *True Positive Rate* (TPR) yang didapatkan dengan cara membagi hasil prediksi positif (dalam hal ini positif *fraud* dan positif *semi fraud*) dengan frekuensi total kondisi positif. Spesifisitas merupakan *True Negative Rate* (TNR) yang didapatkan dengan cara membagi hasil prediksi negatif (negatif *fraud* atau tidak *fraud*) dengan frekuensi total kondisi negatif.

V. KESIMPULAN DAN SARAN

Metode Model Markov Tersembunyi dapat melakukan prediksi status tiap kasus dengan nilai kesalahan prediksi masing-masing sebanyak 1 data pada *semi-positive* (semi *fraud*) maupun *negative* (tidak *fraud*). Model prediksi yang dilakukan metode ini dengan melakukan modifikasi terhadap parameter pada saat penggunaan algoritma Baum-Welch, dan pengaturan ulang terhadap atribut PBF yang digunakan. Penggunaan algoritma Viterbi dimanfaatkan untuk mencari *maximum path* dari tiap kasus, yang dimultiplikasikan hingga akhir untuk mendapatkan kelas kasus yang dihitung.

Dengan hasil yang telah dicapai, maka saran yang dapat diberikan setelah penyelesaian penelitian ini ialah penggunaan Model Markov Tersembunyi Berganda untuk optimasi sensitivitas maupun spesifisitas deteksi, dengan variasi atribut yang lebih beragam, dan pemberian informasi masing-masing atribut secara jelas, menurut kelas riilnya.

VI. DAFTAR PUSTAKA

- [1] Z. Ghahramani, "An Introduction to Hidden Markov Models and Bayesian Networks," *International Journal of Pattern Recognition and Artificial Intelligence*. 15(1):9-42, pp. 9-42, 2001.

- [2] S. Huda, R. Sarno dan T. Ahmad, "Fuzzy MADM Approach for Rating of Process-Based Fraud," *Journal of ICT Research and Applications, Institut Teknologi Bandung*, vol. 9, no. 2, pp. 111-128, 2015.
- [3] Y. A. Effendi, F. Haryadita dan R. Sarno, "Modified Time-Based Heuristics Miner for Parallel Business Processes," *International Review on Computers and Software (IRECOS)*, vol. 11, no. 3, p. 249, 2016.
- [4] R. Sarno, R. D. Dewandono, T. Ahmad, M. F. Naufal dan F. Sinaga, "Hybrid Association Rule Learning and Process Mining for Fraud Detection," *IAENG International Journal of Computer Science*, 42:2, vol. 42, no. 2, pp. 59-72, 2015.
- [5] K. R. Sungkono dan R. Sarno, "Hidden Markov Model for Process Mining of Parallel Business Processes," *International Review on Computers and Software (IRECOS)*, vol. 11, no. 4, pp. 290-300, 2016.
- [6] L. S. Moss, "Example of The Baum-Welch Algorithm," Indiana University, Bloomington, Spring 2008.
- [7] G. A. Fink, *Markov Models for Pattern Recognition: From Theory to Applications*, Dortmund: Springer, 2008.
- [8] K. R. Sungkono dan R. Sarno, "Coupled Hidden Markov Model for Process Mining of Invisible Prime Tasks," *International Review on Computers and Software (IRECOS)*, vol. 11, no. 6, pp. 539-547, 2016.
- [9] D. Jurafsky dan J. H. Martin, "Speech and Language Processing," dalam *Chapter 8: Hidden Markov Model*, Prentice Hall, 2014, pp. 1-21.
- [10] A. Tenyakov, "Estimation of Hidden Markov Models and Their Applications in Finance," Electronic Thesis and Dissertation Repository, Western University, Ontario, 2014.
- [11] E. Britannica, "Markov Process," Encyclopædia Britannica Online, 2016. [Online]. Available: <http://www.britannica.com/topic/Markov-process>. [Diakses 12 Juni 2016].